

GetFit – Fitness App

Mobile Application Programming – INFT2051

Author: Sharjeel Sohail (3316130)

Objective:

The main objective of the report is to explain the project that I have been working on and what functionalities and features it includes and how it solves a real-world problem and connects with the users. It also explains what problems I experienced and how I possibly managed to fix it throughout the development process, and what problems and features the application still lacks.

Purpose:

This app – GetFit, deployed on Android, is designed and developed to solve a real-world issue, which is maintaining health. Who is not talking about fitness and health these days? Everyone is trying somehow to get into better shape and be fit and trust me, smart phones have hell of a solution to it. You don't necessarily have to pay your hard-earned money to gym and fitness centres, your mobile phone would do the work for you if you know how to use it properly.

Since most of us nowadays carry our cell phones with us all time all day, GetFit app is developed to track your steps while you walk, run, or even when you wake up in the middle of the night to eat that last slice of pizza you left the day before. While the application tracks your steps when you walk, it also lets a user keep track of their diet and workout plans by making notes in specified entry fields. This feature can help you keep record of your meals and preparation of each day and help you prepare and plan better for the following days.

Features Included:

The application uses a couple of features which almost every smartphone has these days;

1) Accelerometer (Motion Detection)

This functionality is used to count steps by measuring the acceleration of motion of a structure, in our case, a phone. We simply use the Pythagorean theorem to calculate the magnitude of the acceleration vector of each sample from the accelerometer by taking three axis values; X, Y and Z and then checks the difference between the last magnitude calculated and latest magnitude counted. The scientists before us have already discovered the MET values for us which tells us if the difference is more than 4/5, means that the person is walking and if it's more than 7/8, it's probably because the person is running. However, no one ever yet calculated the exact steps but we can only get the closest steps covered, and that too, by taking quite a few things into consideration i.e. height and weight of a person, which unfortunately I could not cover given the time constraint. However, I developed a basic algorithm using MET values and not considering individuals specification into this.

2) User Input

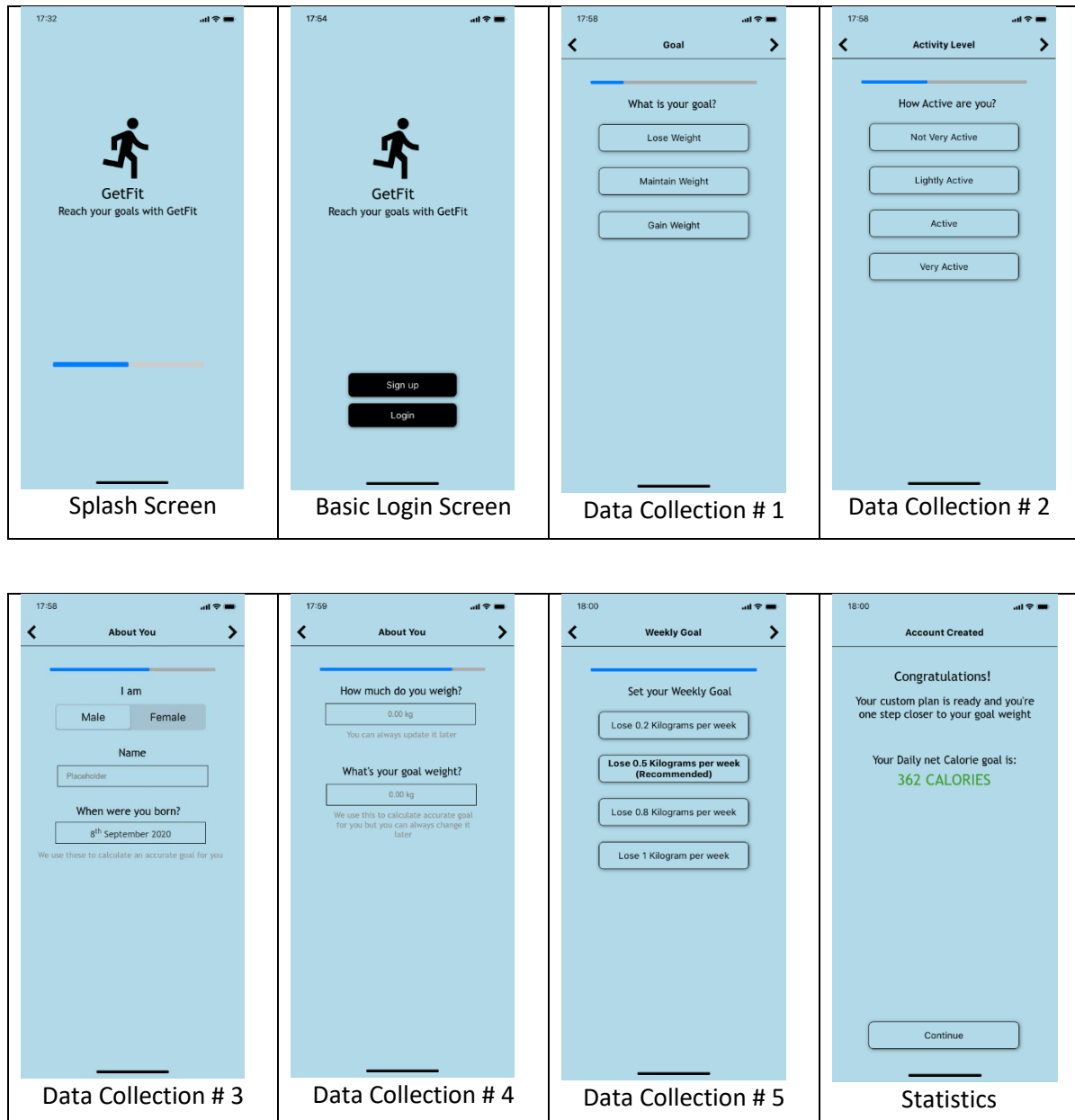
We have minimal User Input that requires consumer to set some basic information such as their name and their current and goal weight just to keep track of where they started and how far they are. They can also enter their daily fitness routine if they choose to, data like their meals during the day and what workout they are planning to do each day, and of course you can save that and check later based on your preferences.

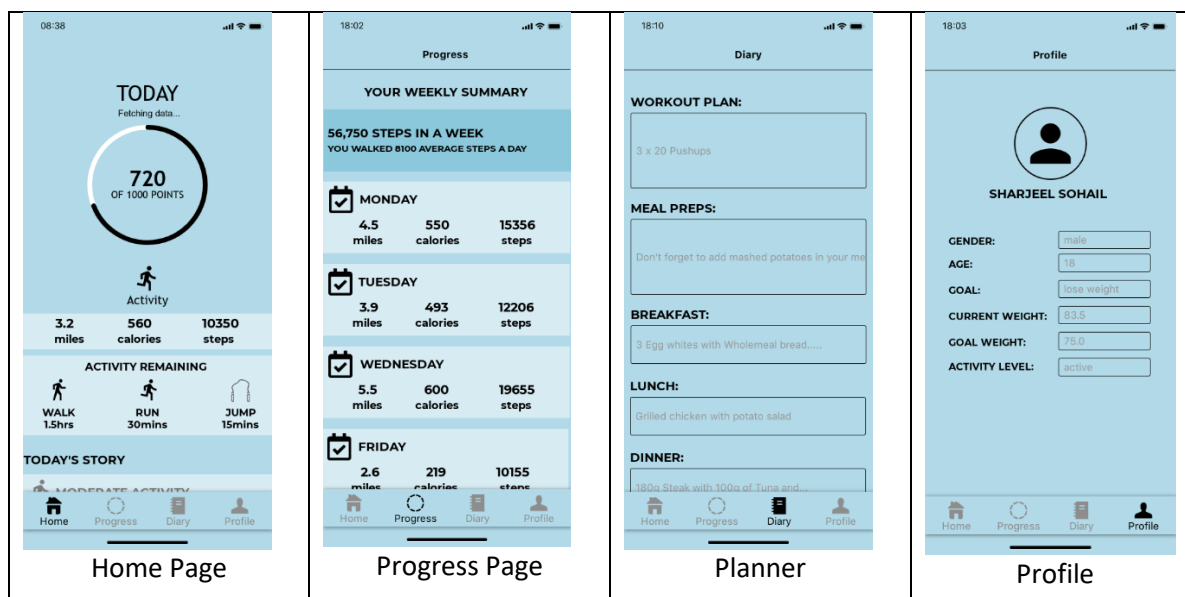
3) Local Database

The default location that GetFit uses is the local folder in the phone to save your daily planning notes and your personal data.

Planning:

The planning of the application was done earlier this semester. I will take you through the initial prototype and explain as I go





I planned on completing a lot of data collection pages and then moving towards a tabbed page which has the functionality and usage. First page shows the steps and related stats. Progress page was planned to show the weekly statistic divided into individual days of the week. Then you have your daily planner and lastly you have your profile which shows your data and allows you to update it.

Completion:

I managed to achieve most of it which I had planned including Home page, planner, profile from the initial design as well as ended up adding few more pages.

The algorithm that I have used to count the steps is basic and not a complex one. I'm taking the acceleration values of X, Y, and Z axis of the phone and calculating the magnitude with those values and then by checking the difference between the last magnitude recorded and the current magnitude value, and finally comparing those values with MET values of 4 and 6 for walking and running respectively.

Planner is being done by using local storage of the phone, it saves all your notes and in a text file and displays it in a listView nicely organised by latest date. You can also modify or update those planner notes and if you like to, as well as delete those files when needed.

Unfinished:

Couple of unexpected bugs and errors I ran into while completing my project as well as given the time constraint it looked quite difficult to complete by the submission date. I skipped few things that I intended to complete, I unfortunately have to write those things in a list which I know is a shame;

- 1) Data Collection Pages – Initially I planned on adding those pages at first and then move to the main application (tabbed page) but after doing some research, I realised that Xamarin doesn't allow you to go from navigation page to a tabbed page, and as I used tabbed page for most of my pages, I chose not to go back change the whole UI just for this. However, what I managed to do another navigation page inside tabbed page which lets you enter your data.

- 2) Counting miles – I also intended to count miles while running but due to the given time constraint I ran out of time and couldn't use GPS to collect location data and apply that into my app, so I decided to remove those UI elements that I had first developed in my app.
- 3) Presenting Data statistics broken down in individual days – Again, time constraint was a big issue for me unfortunately and could not cover this bit as looks quite a work which I did not realise while designing it.

All those things that I could not complete, most of them were because of the time limitation. However, I'm looking forward to implement those features in the future to learn and increase my skills and finish what I started.

References:

- 1) The idea of accelerometer algorithm is being taken from a YouTube video. The video shows the code using Android Studio <https://www.youtube.com/watch?v=o-qpVefrfVA>
- 2) Accessing accelerometer in Android MainActivity.cs and HomePage.xaml.cs is taken from Microsoft video which shows how to get the XYZ reading and update when readings changed <https://www.youtube.com/watch?v=h2MGTh5bkdA>
- 3) The idea of reading and writing to files was taken from GitHub but was completely understood and changed by myself and little help from Hayden to fix few bugs <https://github.com/xamarin/xamarin-forms-samples/tree/master/GetStarted/Notes/MultiPage>

Check ReadMe file for testing instructions